International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR)
ISSN(P): 2249-6831; ISSN(E): 2249-7943
Vol. 7, Issue 2, Apr 2017, 25-28

TRANS STELLAR
•Journal Publications •Research Consultancy

AN EFFICIENT RESOURCE SCHEDULING ALGORITHM USING KNAPSACK

RASHMI KUMARI[1] & RAKSHA PANDEY[2]

[1]*Department of Computer Science and Engineering, Institute of Technology, Guru Ghasidas Viswavidyalaya, Bilaspur, Chhattisgarh, India*

[2]*Assistant Professor, Department of Computer Science and Engineering, Institute of Technology, Guru Ghasidas Viswavidyalaya, Bilaspur, Chhattisgarh, India*

ABSTRACT

Grid computing is utilized in variety of computational areas now-a-days. It is the next generation of distributed computing. In grid computing, we try to integrate many heterogonous, geographically distributed computing systems into one. All the systems in the grid are linked together to provide better resources. Hence, we can use resources of any system, which is in grid, for our task. This paper presents a job scheduling algorithm that can utilize the resources in efficient way.

KEYWORDS: Grid Computing, Job Scheduling, Knapsack & Scheduling Algorithm

Received: Jan 27, 2017; **Accepted:** Mar 09, 2017; **Published:** Mar 14, 2017; **Paper Id.:** IJCSEITRAPR20174

Original Article

INTRODUCTION

GRID computing comes under virtual computing, i.e. we can choose resource in some manner and allocate jobs on it. Grid allows resource sharing on a large scale. Thus providing high speed execution of jobs. Grid has two major tasks i.e. resource allocation and job scheduling. Job scheduling and proper resource allocation plays a vital role in improvement of efficiency and proficiency of resources. This is the world of advanced computing in which Personal Computers can also perform some advanced task, But still there is need of a strong Environment where we can perform the tasks that requires alots of resources and Grid computing is the good solution for the problem.[2]. Resources of idle systems are utilised properly. Job scheduling is done to maximize throughput, minimise average running time, increase efficiency and decrease average waiting time.

In this paper we try propose an algorithm that will increase efficiency of resources. We will compare our algorithm with some trivial existing algorithms. Process scheduling is essential part of operating system. The goal of this paper is to find optimal scheduling algorithm for efficient resource utilisation. This paper is framed as follows: section II contains related works. Section III includes proposed algorithm. Section IV constitutes Experimental Evaluation. Finally section V covers conclusions and lastly it has references.

RELATED WORKS

Resource management and job scheduling are two major challenging tasks in grid computing. Various works has been done in this area. Simplest algorithm is first come first serve (FCFS) algorithm where jobs are scheduled as per their arrival time. Another algorithm is shortest job first (SJF) which is similar to FCFS but instead of selecting jobs on the basis of arrival time, we select them on the basis of their execution time. As the name suggests jobs having small value for execution time are executed first. In priority based scheduling, Priority

is assigned to every process. Process with higher priority is executed first. SJF based priority scheduling algorithm[3] not only reduces the average waiting time of jobs but also reduces average turn around Time of Jobs. In HCCG[4], Comparison is done on the basis of different constraints as Processing cost, File Size and Bandwidth but the main focus was on Resource's compaction degree. In GBJS[5] basic comparision criteria was Memory size and algorithm shows better result comparatively to some existing algorithms. DJGBSDA[6]focuses on Resource's Computational Power and it was not only reduces Processing cost but also Jobs Communication Time. In SBJGBS [7] two basic criteria's Bandwidth and Resource capability for computation is taken. BAJGBS[8] also uses the same parameter as Bandwidth and Resource's Computational Power but added a new criteria of highest Communication and Transmission rate of Bandwidth. GBFJS[9] having disadvantage as higher time Complexity, Higher Pre-processing time for scheduling of jobs. And finally it does not pay any attention of the memory file-size. All these algorithm focuses on resource utilization, we proposed our algorithm to minimize the processing cost with efficient resource utilization.

PROPOSED ALGORITHM

This algorithm is similar to backtracking method for knapsack problem. First processor is considered as the knapsack. First max heap tree is constructed on the basis of execution time of each process i.e. in accordance with the longest job first then we will move from root node to left nodes. Each job is assigned to a resource. If it's not then we will backtrack the path and move to the right node.

Step 1: Set Total_Time = Sumof execution times of all the processes.

Step 2: Set Average_time = Total_Time/Number of processors.

Step 3: Sort the processes as their execution time in decreasing order.

Step 4: For i = 0 to number of processors repeat from step 5to step 9.

Step 5: Set Total_Runtime = 0 and Max_Process_Time = 0

Step 6: For j = 0 to number of processes repeat from step 7and step 8.

Step 7: Check if Total_Runtime + Pj. Exceution Time<= Average_Time

Step 8: Do{

Execute process Pj

Update Total_Runtime = Total_Runtime+Pj.ExecutionTime

}

Step 9: If Total_Runtime > Max_Process_Time Then

Max_Process_Time = Total_Runtime

Step 10: If any job is not executed then execute it at the end as per the FCFS scheduling algorithm.

EXPERIMENTAL EVALUATION

Experimental Setup and Comparison

For performance analysis, we create Simulation Environment using GridSim simulation toolkit [10]. A simulation

is performed using various characteristic in heterogeneous environment of GridSim toolkit. Our focus was on processing cost and we analyses the performance of proposed algorithm with AFJS[11], HCCG[4] and DJGBSDA[6].

Table 1: Comparison between Proposed with HCCG, AFJS and DJGBSDA

Number of Jobs	Processing Time			
	Knapsack	HCCG	AFJS	DJGBSDA
100	71	48	81	92
200	72	93	168	195
300	72	165	234	254
400	68	222	335	476
500	72	264	396	529

Figure 1

CONCLUSIONS

New Proposed scheduling strategy more utilizes the resource and it reduces total processing cost. The simulation result clearly shown that 'an efficient resource scheduling algorithm using knapsack problem' reduces processing cost better than HCCG[4], AFJS[11] and DJGBSDA[6]. According to the Simulation result, 'an efficient resource scheduling algorithm using knapsack problem' performs much better as the number of jobs increases.

REFERENCES

1. *Foster, I. and Kesselman, C. Computation Grids. Foster, I. and Kesselman, C. eds. The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 1999, 2-48.*
2. *Kleinrock, L. MEMO on Grid Computing, University of California, Los Angeles, 1969.*
3. *Rukhsar Khan et al, International Journal of Computer Science and Mobile Computing, Vol.4 Issue.9, September -2015, pg. 324-331*
4. *Abhinav Srivastava, Rituraj Rathore & Raksha Sharma, HIGH COMPACTION COARSE GRAINED JOB SCHEDULING IN GRID COMPUTING, International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR) ISSN 2249-6831 Vol. 3, Issue 2, Jun 2013, 295-302*
5. *Vishnu Kant Soni, Raksha Sharma, Manoj Kumar Mishra, "Grouping-Based Job Scheduling Model in Grid Computing", World Academy of Science, Engineering and Technology 41 2010.*
6. *N. Muthuvelu, Junyan Liu, N. L. Soe, S. Venugopal, A. Sulistio, and R. Buyya, "A dynamic job grouping-based scheduling for deploying applications with fine-grained tasks on global grids," in Proc of Australasian workshop on grid computing, vol. 4, pp. 41–48, 2005*
7. *Ng Wai Keat, Ang Tan Fong, "SCHEDULING FRAMEWORK FOR BANDWIDTH-AWARE JOB GROUPING- BASED SCHEDULING IN GRID COMPUTING", Malaysian Journal of Computer Science, Vol. 19, No. 2, pp. 117-126, 2006.*
8. *T. F Ang, W. K. Ng, T. C Ling, "A Bandwidth-Aware Job Grouping-Based Scheduling on Grid Environment", Information Technology Journal, Vol.8, NO.3, pp. 372-377, 2009.*
9. *Quan Liu, Yeqing Liao, "Grouping-based Fine-grained Job Scheduling in Grid Computing", IEEE First International Workshop on Educational technology And Computer Science, Vol.1, pp. 556-559, 2009.*
10. *Buyya, R. and M. Murshed, 2002. GridSim: A toolkit for the modeling and simulation of distributed management and scheduling for grid computing.,Concurrency Computat.: Pract. Exper.2002;14:1175–1220 (DOI: 10.1002/cpe.710)*
11. *Yeqing Liao, Quan Liu, "Research on Fine-grained Job scheduling in Grid Computing", I.J. Information Engineering and Electronic Business, 2009, 1, 9-16 Published Online October 2009 in MECS (http://www.mecs-press.org/)*